Lecture 4 – Message authentication, UF-CMA, variable-input-length PRFs, CMAC

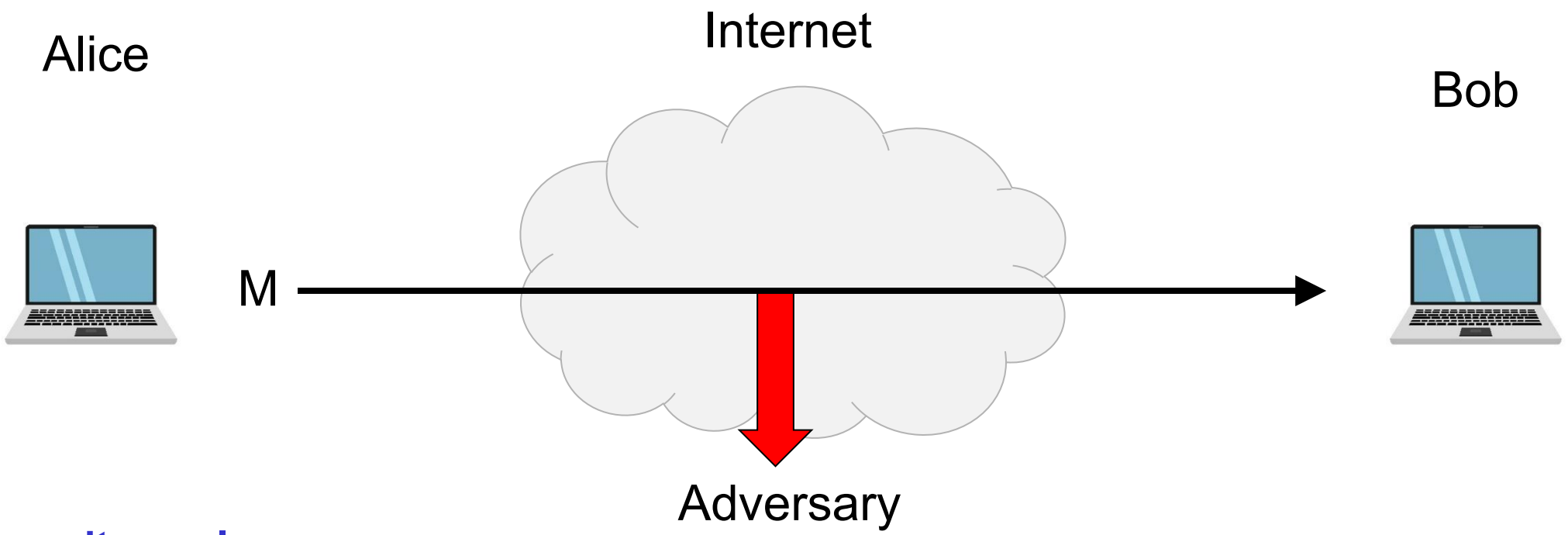
TEK4500

09.09.2026

Håkon Jacobsen

hakon.jacobsen@its.uio.no

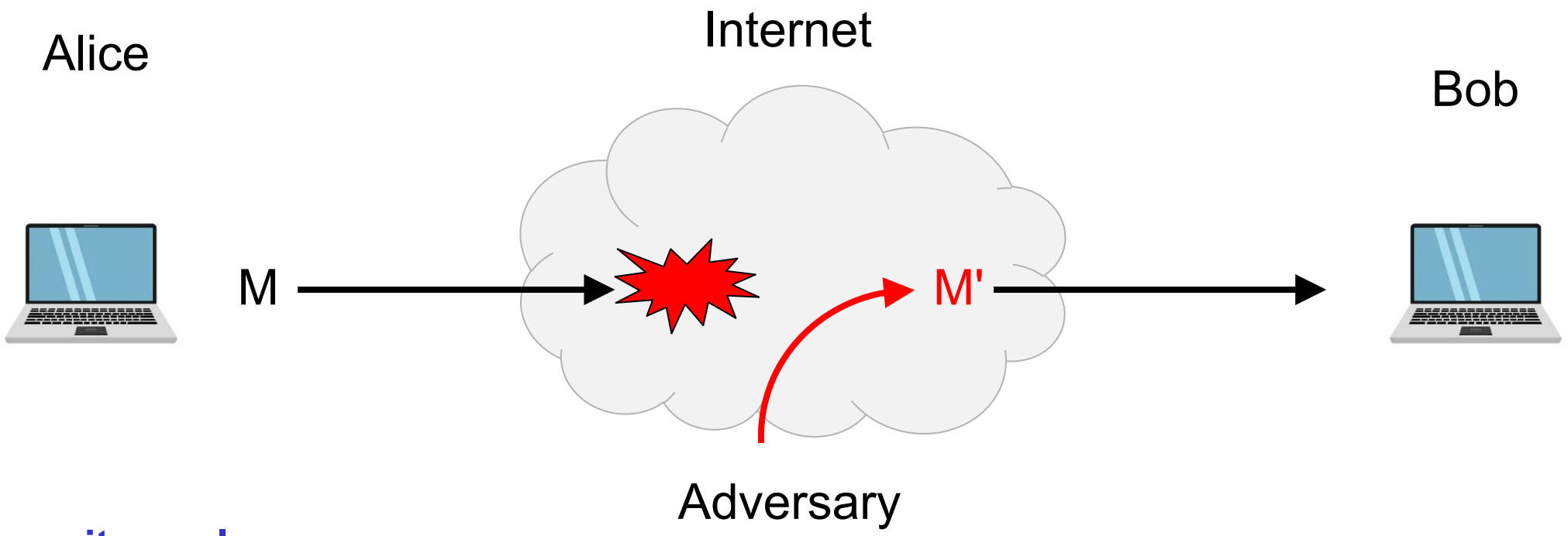
What is cryptography?



Security goals:

- **Data privacy:** adversary should not be able to read message M
- **Data integrity:** adversary should not be able to modify message M
- **Data authenticity:** message M really originated from Alice

What is cryptography?



Security goals:

- **Data privacy:** adversary should not be able to read message M
- **Data integrity:** adversary should not be able to modify message M
- **Data authenticity:** message M really originated from Alice

Basic goals of cryptography

	Message privacy	**Message integrity / authentication**
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption)	Digital signatures

Motivation

- **Goal:** *integrity*, not privacy

- **Examples:**
  - Protecting OS system files against tampering
  - Browser cookies stored by web servers
  - Control signals in network management

Encryption ≠ integrity

CTR. Enc(K,·)



0110100 1110 100101100110

⊕

"Send Bob $10"

1001010 1010 000000001010

1111110 0100 ~~100~~0110 01101100



CTR. Dec(K,·)



1111110 0100 0110 01101100

⊕

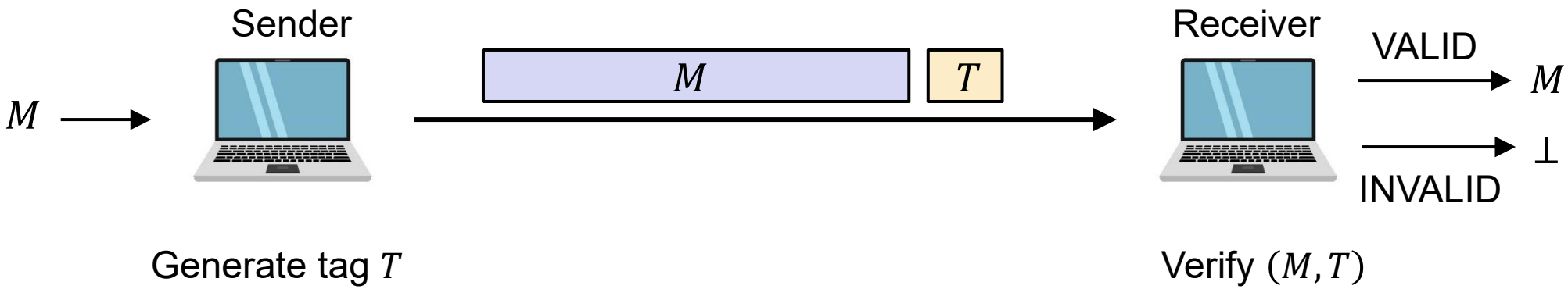
0110100 1110 100101100110

=

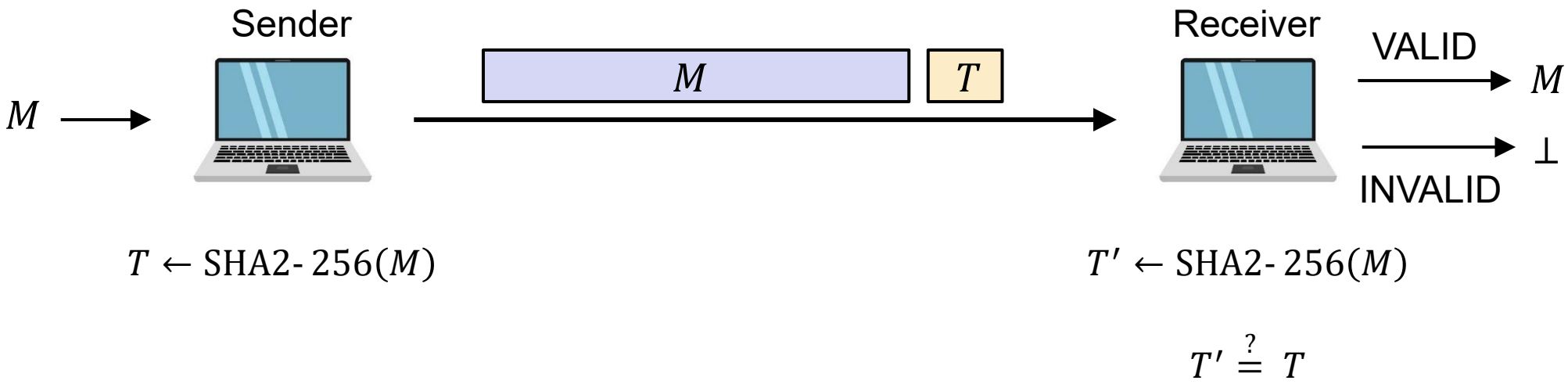
1001010 1010 111 00001010

"Send Bob **$3850"**

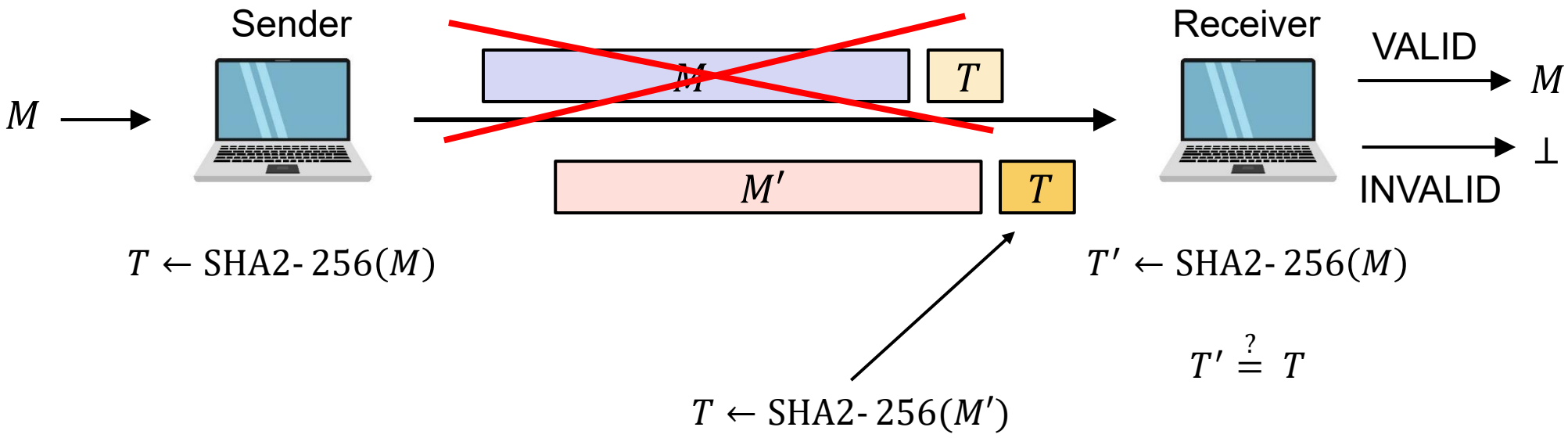
Message authentication – idea



Message authentication – idea



Message authentication – idea



Message authentication codes – syntax

A **message authentication code** is a triple $\Sigma = (\mathrm{KeyGen}, \mathrm{Tag}, \mathrm{Vrfy})$ of algorithms:

$\mathrm{Tag} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$

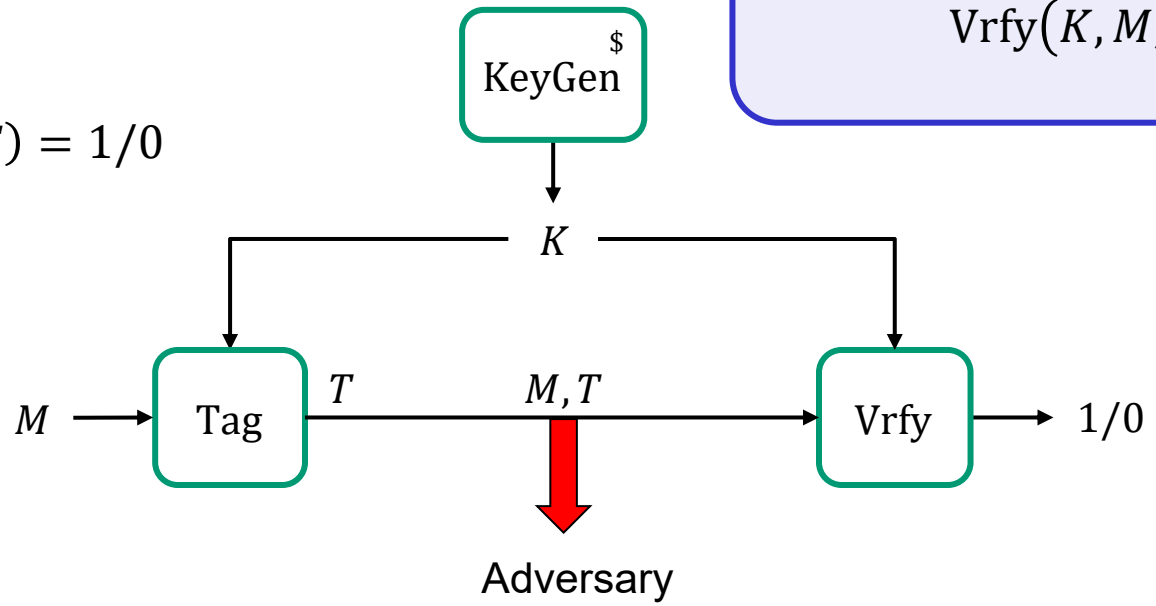
$\mathrm{Tag}(K, M) = \mathrm{Tag}_K(M) = T$

$\mathrm{Vrfy} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0,1\}$

$\mathrm{Vrfy}(K, M, T) = \mathrm{Vrfy}_K(M, T) = 1/0$

Correctness: $\forall K \xleftarrow{} \mathrm{KeyGen}$ and $\forall M \in \mathcal{M}$:

$$\mathrm{Vrfy}\big(K, M, \mathrm{Tag}(K, M)\big) = 1$$



Integrity security goal:

Hard to **forge** a T for any M after first seeing valid tags on $M_1, M_2, \ldots$

UF-CMA – Unforgability against chosen-message attacks

$\mathbf{Exp}^{\text{uf-cma}}_{\Sigma}(A)$

1. $K \twoheadleftarrow \Sigma.\text{KeyGen}$
2. $\mathcal{S} \leftarrow [\,]$
3. forge $\leftarrow 0$
4. $A^{\text{TAG}(\cdot),\,\text{VF}(\cdot,\cdot)}$
5. **return** forge

$\text{TAG}(M)$

---

1. $T \leftarrow \Sigma.\text{Tag}(K, M)$
2. $\mathcal{S}.\text{add}(M)$
3. **return** T

$\text{VF}(M, T)$

---

1. $d \leftarrow \Sigma.\text{Vrfy}(K, M, T)$
2. **if** $d = 1$ **and** $M \notin \mathcal{S}$ **then:**
3. forge $\leftarrow 1$
4. **return** d



Challenger

$\text{Tag}(K,\cdot)$

$M_1 \quad M_2 \quad \ldots$

$T_1 \quad T_2 \quad \ldots$

$\text{Vrfy}(K,\cdot,\cdot)$

$(M'_1, T'_1) \quad (M'_2, T'_2) \quad \ldots$

$1/0 \qquad 1/0 \quad \ldots$

Adversary *wins* if:

- one (M'_i, T'_i) is *valid*, i.e., $\text{Vrfy}(K, M'_i, T'_i) = 1$
- and M'_i is *fresh*, i.e., $M'_i \notin \{M_1, M_2, \ldots,\}$

Definition: The **UF-CMA advantage** of an adversary A is

$$\mathbf{Adv}^{\text{uf-cma}}_{\Sigma}(A) = \Pr\left[\mathbf{Exp}^{\text{uf-cma}}_{\Sigma}(A) \Rightarrow 1\right]$$

Properties of UF-CMA definition

- UF-CMA security implies:
  - Hard to recover K
  - Hard forge a *specific* message chosen by the adversary
  - Tag lengths must be *long enough*!
    - Suppose $|T| = 10$; then

$$\mathbf{Adv}_{\Sigma}^{\text{uf-cma}}(\text{"Guess } T\text{"}) = \frac{1}{2^{10}} \approx \frac{1}{1000}$$

- Does *not* give protection against **replay attacks**
  - Message counters
  - Nonces
  - Timestamps

$\mathbf{Exp}_{\Sigma}^{\text{uf-cma}}(A)$

1. $K \twoheadleftarrow \Sigma.\text{KeyGen}$
2. $\mathcal{S} \leftarrow [\,]$
3. forge $\leftarrow 0$
4. $A^{\text{TAG}(\cdot),\,\text{VF}(\cdot,\cdot)}$
5. **return** forge

TAG(M)

1. $T \leftarrow \Sigma.\text{Tag}(K, M)$
2. $\mathcal{S}.\text{add}(M)$
3. **return** T

VF(M, T)

1. $d \leftarrow \Sigma.\text{Vrfy}(K, M, T)$
2. **if** $d = 1$ **and** $M \notin \mathcal{S}$ **then:**
3. forge $\leftarrow 1$
4. **return** d

$$\mathbf{Adv}_{\Sigma}^{\text{uf-cma}}(A) = \Pr\left[\mathbf{Exp}_{\Sigma}^{\text{uf-cma}}(A) \Rightarrow 1\right]$$

Verification algorithm is not “needed”

deterministic *function*

$$\mathrm{Tag}_K(M) = T$$

$$\mathrm{Vrfy}_K(M, T) \overset{\mathrm{def}}{=} \left(\mathrm{Tag}_K(M) \overset{?}{=} T\right)$$

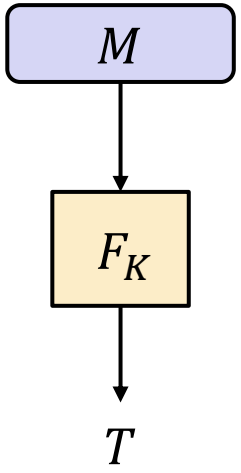
"recompute and check"

- MACs are a special class of message authentication *schemes*
  …but MACs are actually most common in practice

- MAC ⟹ unique tags

PRFs are good MACs

$F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^\tau$



$\Sigma_F.\mathrm{Tag}_K(K, M)$

1. **return** $F_K(M)$

$\Sigma_F.\mathrm{Vrfy}(K, M, T)$

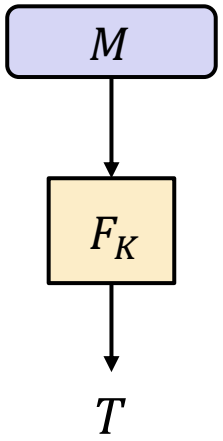
1. **return** $F_K(M) \stackrel{?}{=} T$

Theorem: F is a secure PRF $\Longrightarrow$ Σ_F is a UF-CMA secure MAC (for inputs of length n)

PRFs are good MACs – proof sketch

Theorem: For *any* UF-CMA adversy A, asking v VF queries, there is a PRF-adversary B such that

$$\mathbf{Adv}^{\text{uf-cma}}_{\Sigma_F}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{v}{2^{\tau}}$$

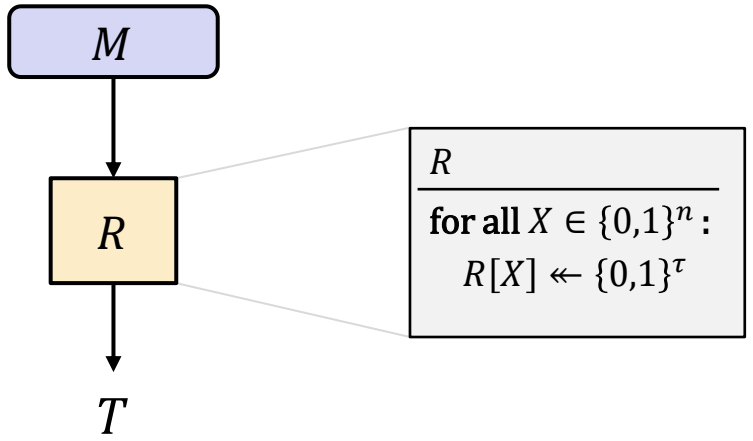


$$\Pr[F_K(M) = T] = ?$$

PRFs are good MACs – proof sketch

Theorem: For *any* UF-CMA adversy A, asking v VF queries, there is a PRF-adversary B such that

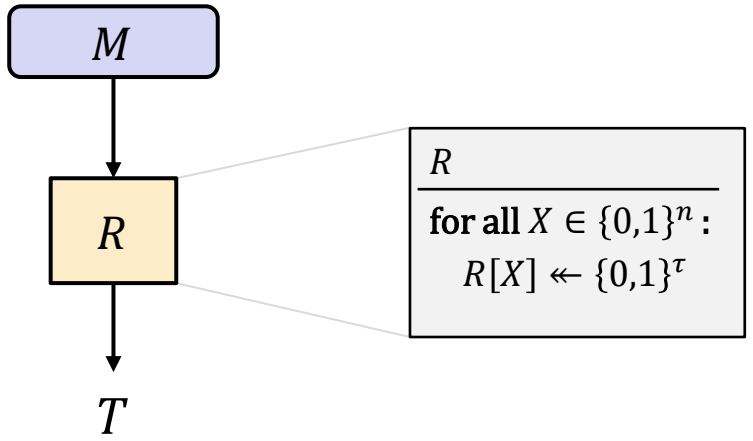
$$\mathbf{Adv}^{\text{uf-cma}}_{\Sigma_F}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{v}{2^{\tau}}$$



PRFs are good MACs – proof sketch

Theorem: For *any* UF-CMA adversy A, asking v VF queries, there is a PRF-adversary B such that

$$\mathbf{Adv}^{\text{uf-cma}}_{\Sigma_F}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{v}{2^{\tau}}$$



$F_K \approx R$ by PRF security

$$\Pr[R[M] = T] = \frac{1}{2^{\tau}}$$

PRFs are good MACs – proof sketch

Theorem: For *any* UF-CMA adversy A, asking v VF queries, there is a PRF-adversary B such that

$$\mathbf{Adv}^{\text{uf-cma}}_{\Sigma_F}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{v}{2^{\tau}}$$



$$\Pr[R[M_1] = T_1 \ \vee \ \cdots \vee R[M_v] = T_v] \ \leq \ \sum_{i=1}^{v} \Pr[R[M_i] = T_i] \ = \ \frac{v}{2^{\tau}}$$

PRFs are good MACs – proof sketch

Theorem: For *any* UF-CMA adversy A, asking v VF queries, there is a PRF-adversary B such that

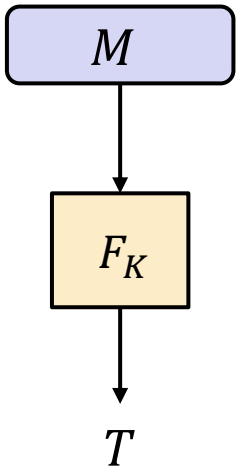
$$\mathbf{Adv}^{\text{uf-cma}}_{\Sigma_F}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{v}{2^{\tau}}$$



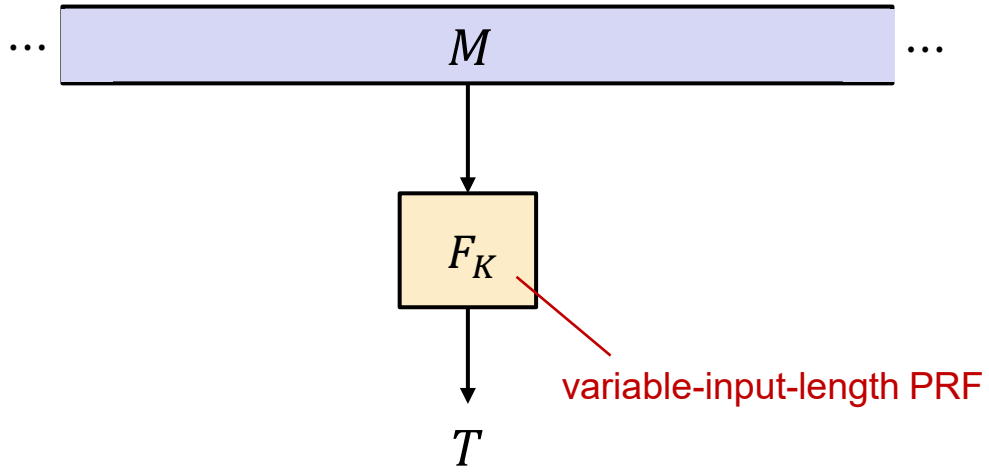
$$\Pr[R[M_1] = T_1 \ \vee \ \cdots \vee R[M_v] = T_v] \ \leq \ \sum_{i=1}^{v} \Pr[R[M_i] = T_i] \ = \ \frac{v}{2^{\tau}}$$

PRFs are good MACs

What we have…



What we want…



Example: $F = \text{AES-256}$ is a good MAC

Variable-input-length PRF

Fixed-input-length PRF

$F : \{0,1\}^k \times \{0,1\}^{in} \to \{0,1\}^{out}$

Variable-input-length PRF

$F : \{0,1\}^k \times \{0,1\}^{\leq in} \to \{0,1\}^{out}$

$\{0,1\}^{\leq in}$ – set of all bitstrings of length $\leq in$

Example: $\{0,1\}^{\leq 3} = \{0,1,00,01,10,11,000,001,\dots,111\}$

$= \{0,1\}^1 \cup \{0,1\}^2 \cup \{0,1\}^3$

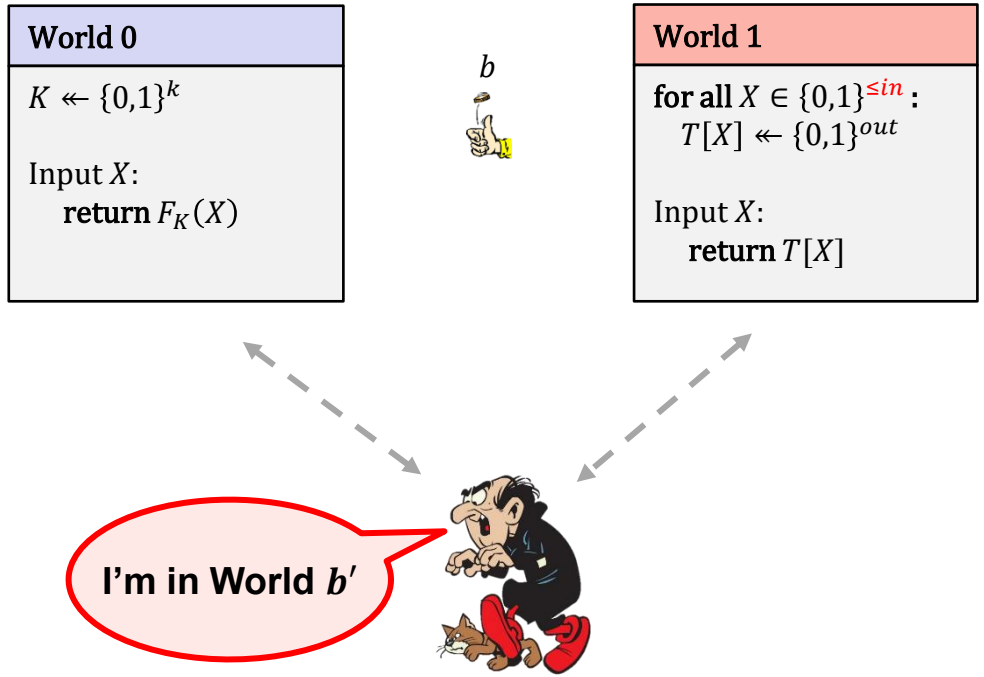
Variable-input-length PRF – security

$\mathbf{Exp}_F^{\text{vil-prf}}(A)$

1. $b \twoheadleftarrow \{0,1\}$
2. $K \twoheadleftarrow \{0,1\}^k$
3. $T \leftarrow$ empty dictionary
4. **for all** $X \in \{0,1\}^{\leq in}$
5. $\quad T[X] \twoheadleftarrow \{0,1\}^{out}$
6.
7. $b' \leftarrow A^{\mathcal{O}(\cdot)}$
8. **return** $b' \stackrel{?}{=} b$

$\mathcal{O}(X)$:

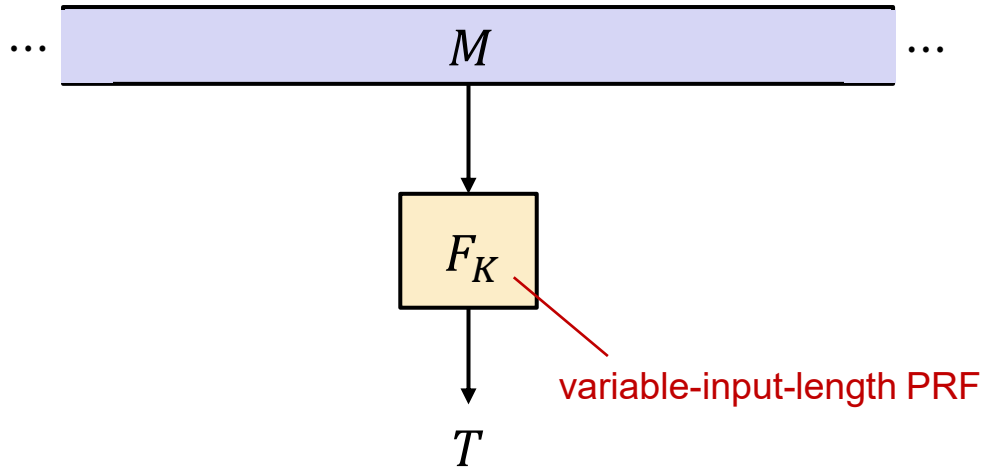
1. **if** $b = 0$:
2. $\quad$ **return** $F_K(X)$
3. **else:**
4. $\quad$ **return** $T[X]$



Definition: The **VIL-PRF-advantage** of an adversary A is

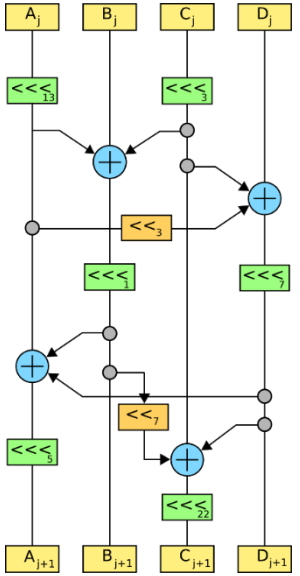
$$\mathbf{Adv}_F^{\text{vil-prf}}(A) \stackrel{\text{def}}{=} 2 \cdot |\Pr[b' = b] - 1/2|$$

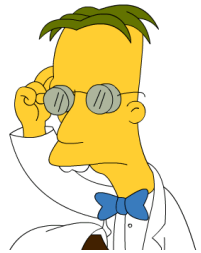
Variable-input-length PRFs are good MACs



Theorem: F is a secure variable-input-length PRF $\Longrightarrow$ F is a UF-CMA secure MAC for *all input lengths*

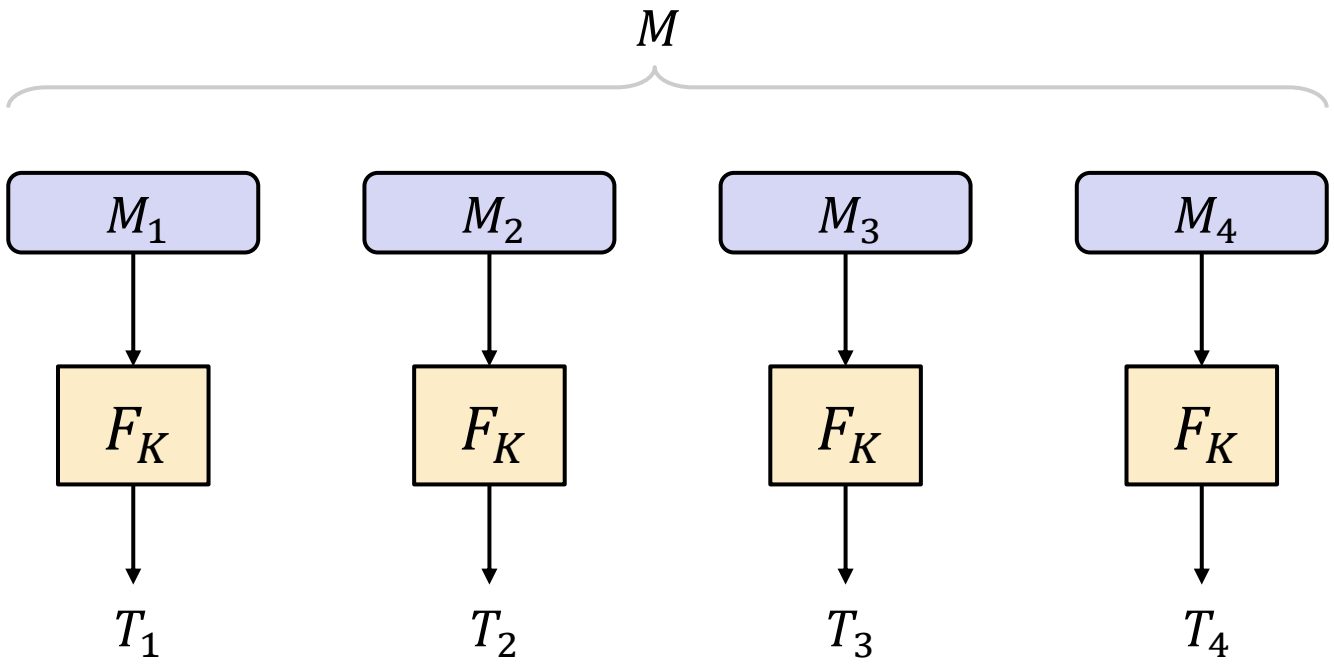
Proof: Identical to the fixed-input-length PRF case





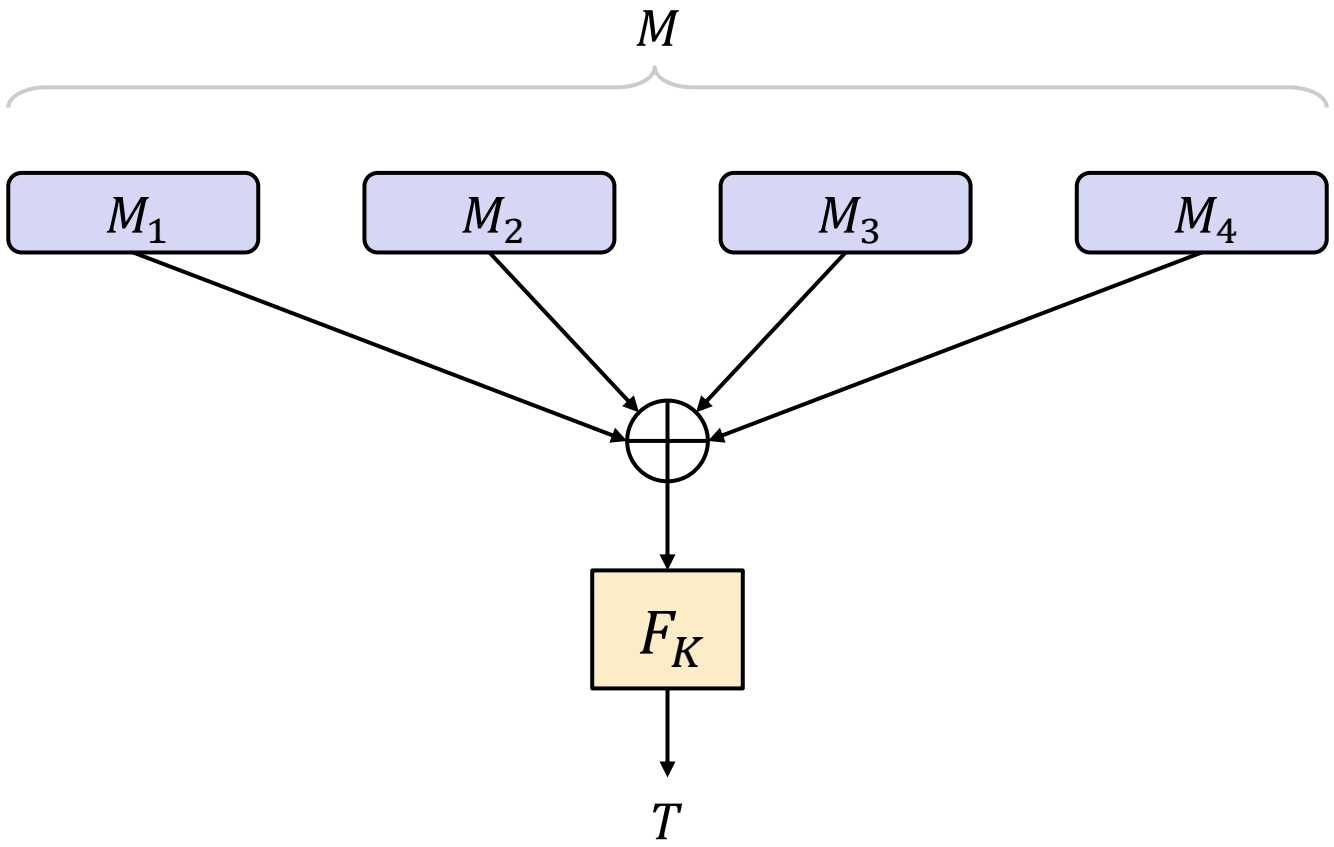
Variable-input-length PRFs

Variable-input-length PRFs – attempt 1

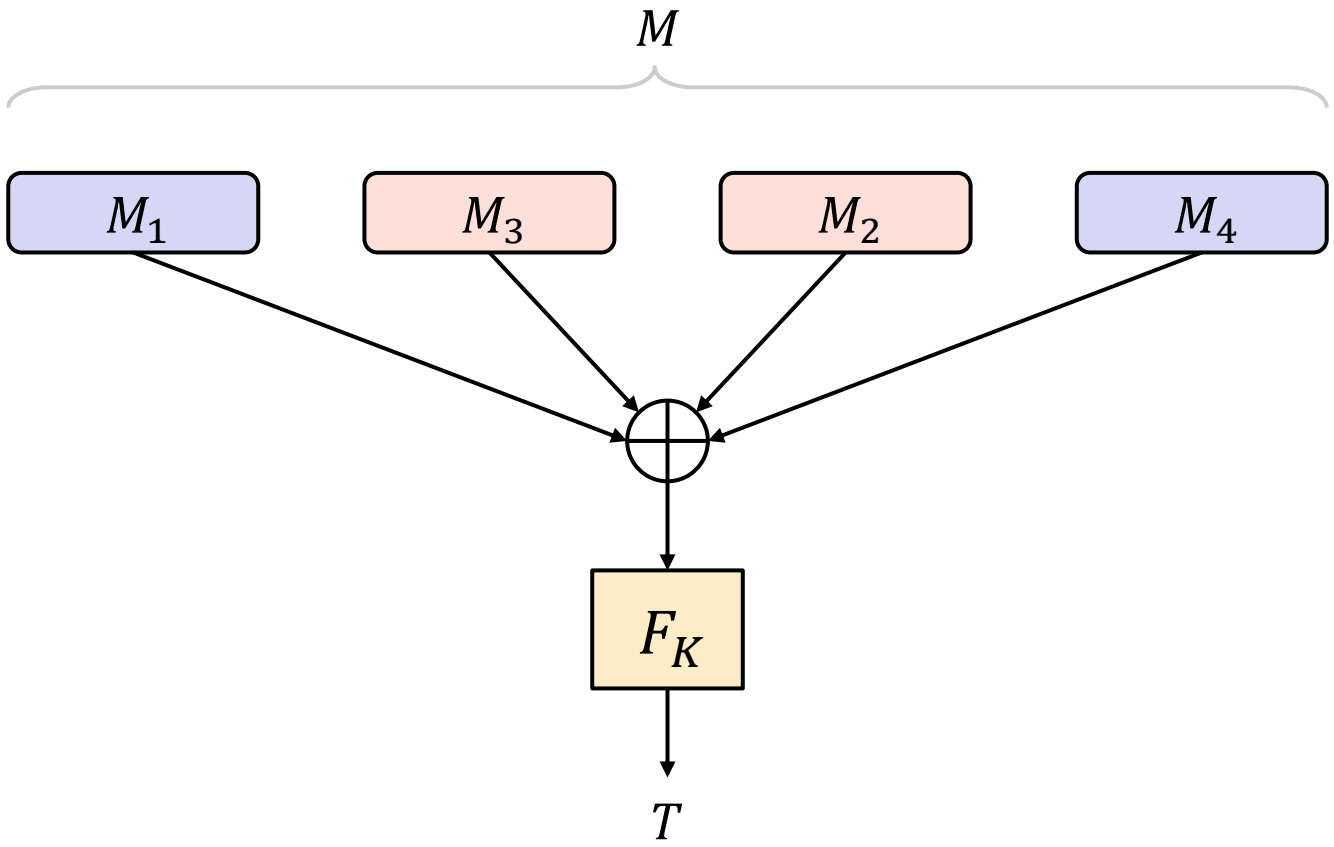


$$T = T_1||T_2||T_3||T_4$$

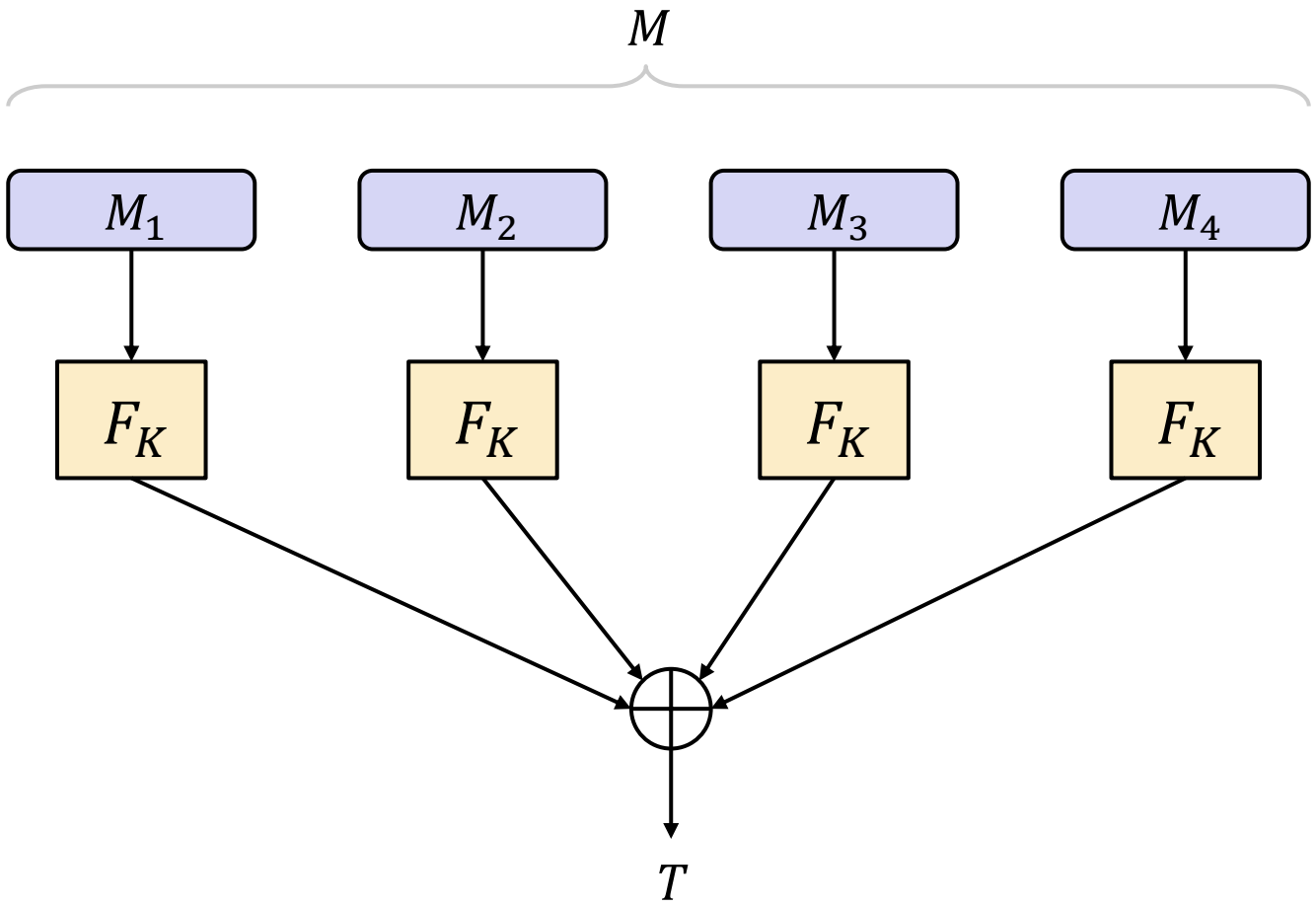
Variable-input-length PRFs – attempt 2



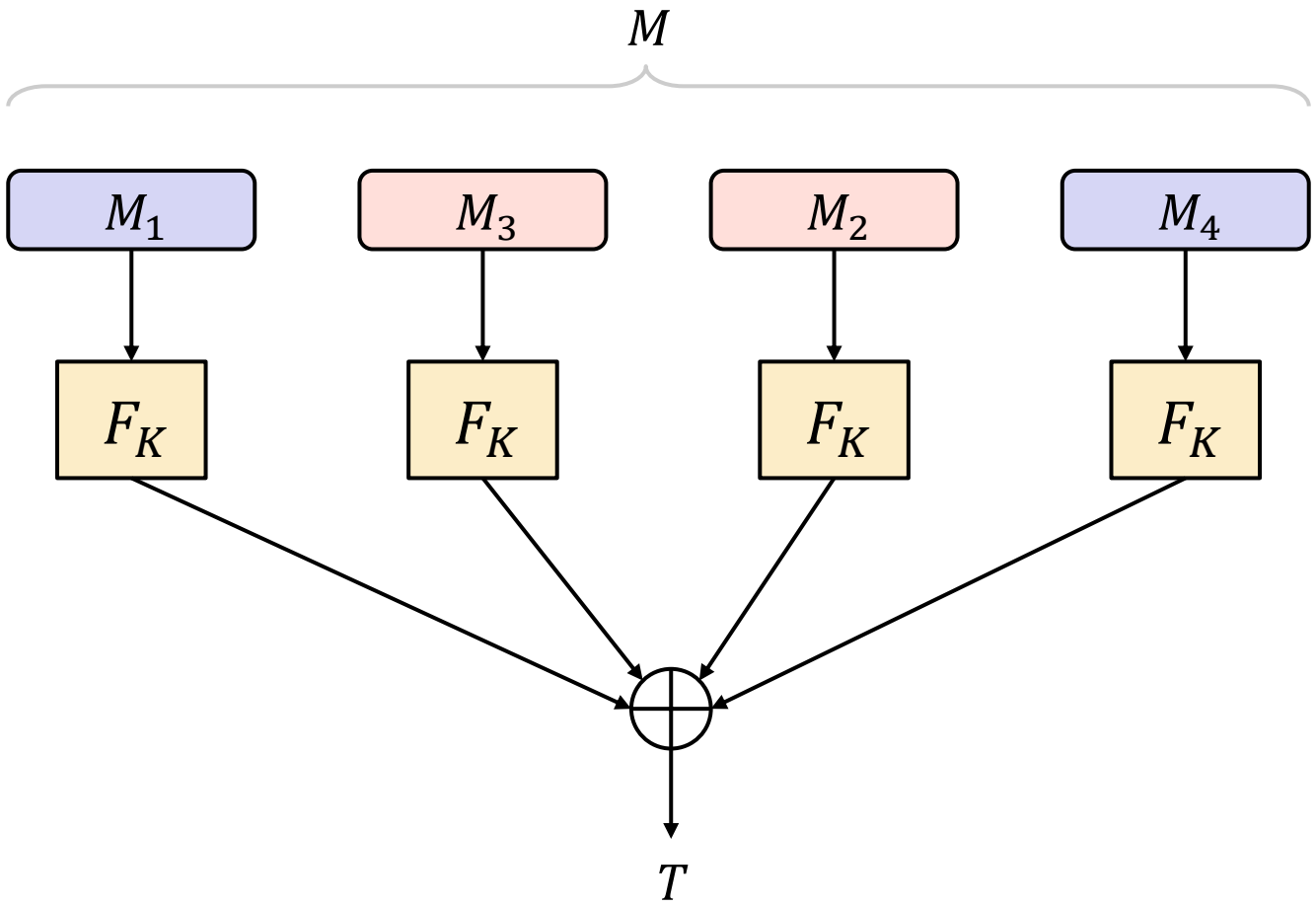
Variable-input-length PRFs – attempt 2



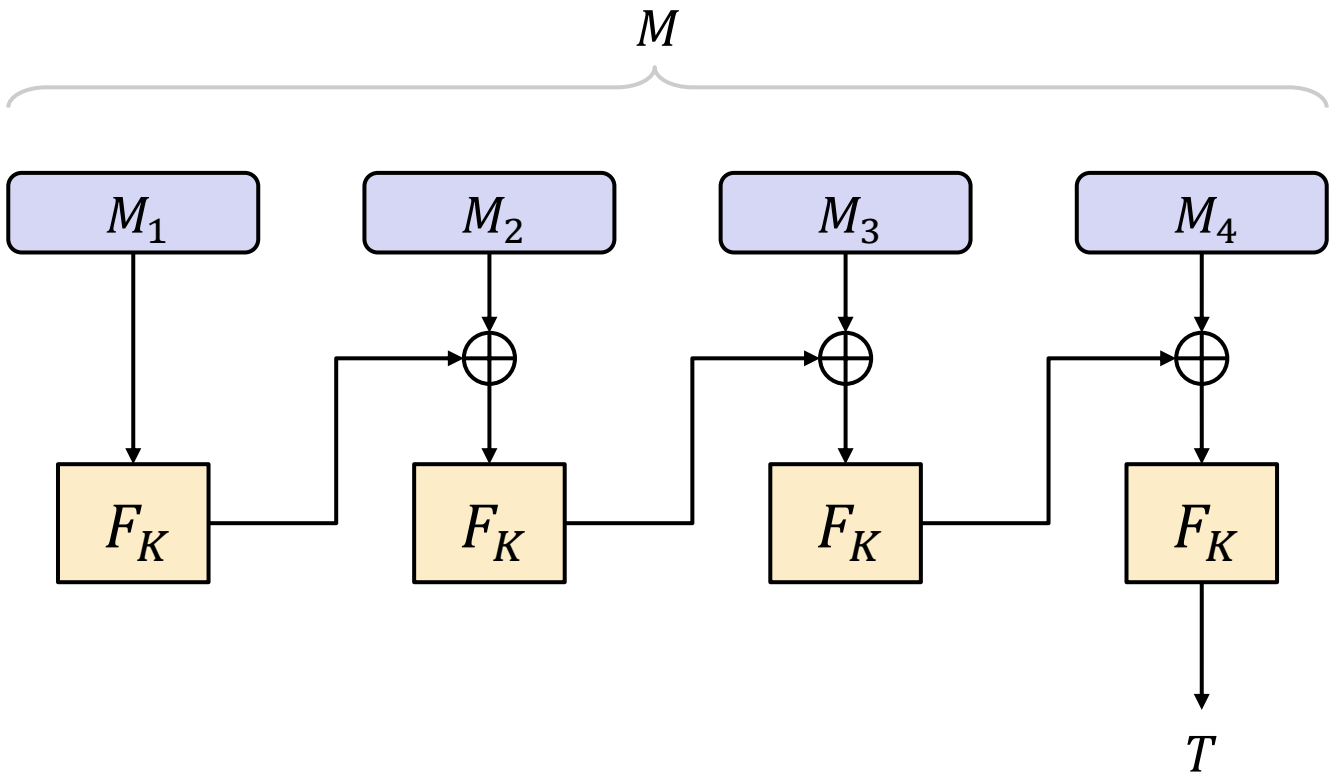
Variable-input-length PRFs – attempt 3



Variable-input-length PRFs – attempt 3



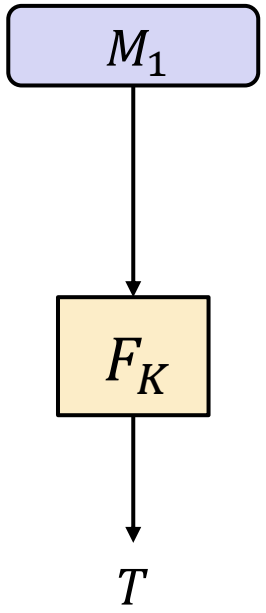
Variable-input-length PRFs – attempt 4 (a.k.a. CBC-MAC)

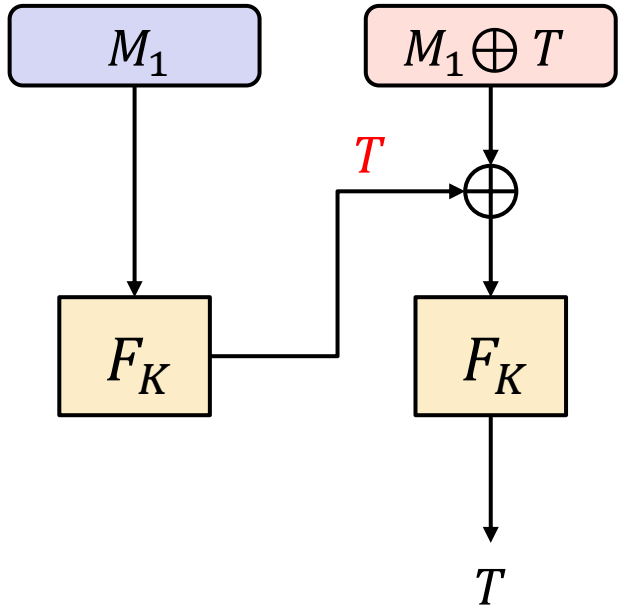


Theorem: F is a secure fixed-input-length PRF $\Longrightarrow$ CBC-MAC is a secure **fixed-input-length** PRF

not what we wanted!

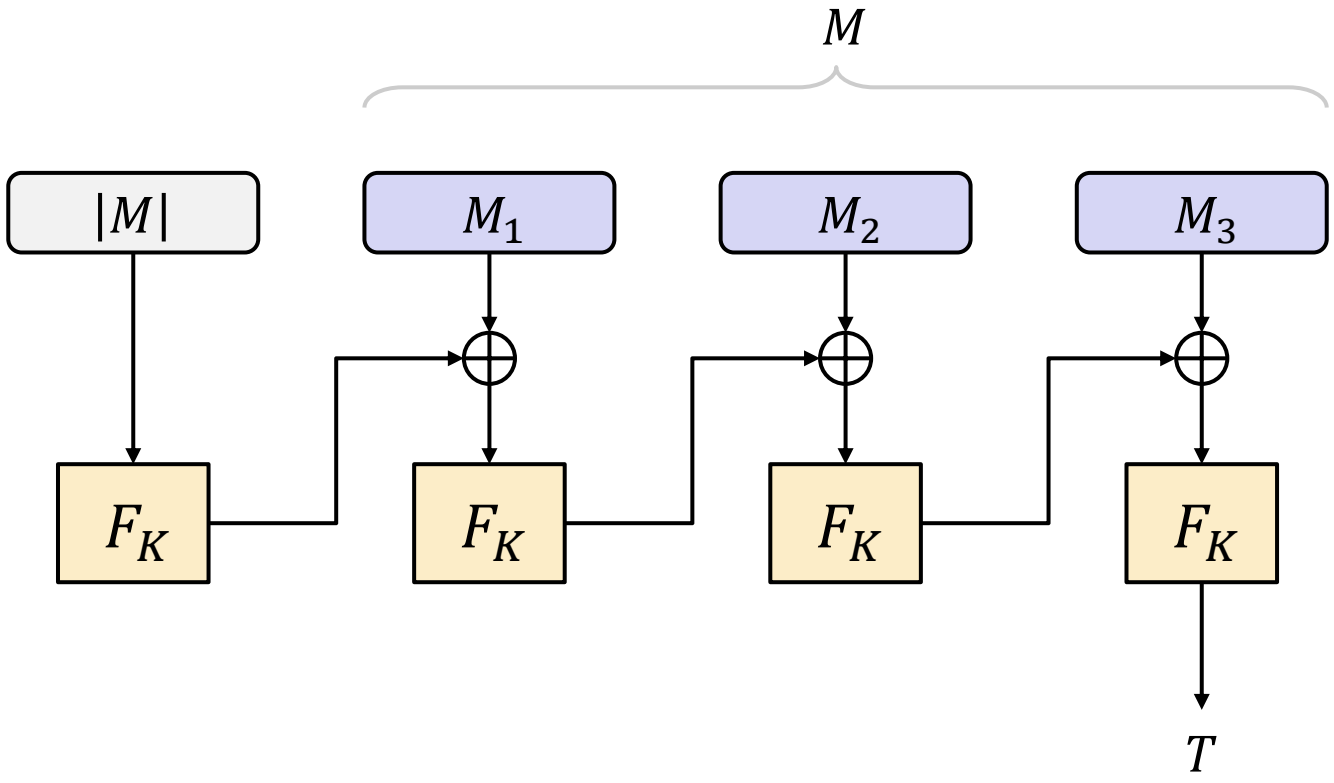
CBC-MAC is not a secure *variable-input-length* PRF





$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{CBC\text{-}MAC}}(A) \approx 1$$

Length-prepended CBC-MAC

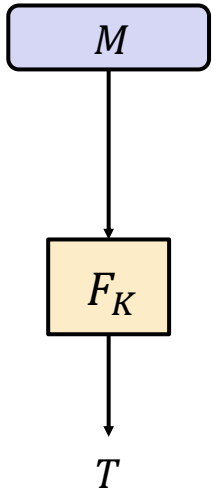


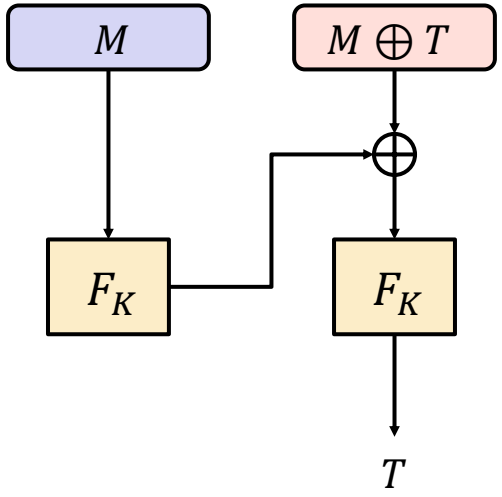
Theorem: F is a secure fixed-input-length PRF $\Longrightarrow$ Length-prepended CBC-MAC is a secure **variable-input-length** PRF

Question: What if you instead add the length to the end?

Length-prepended CBC-MAC

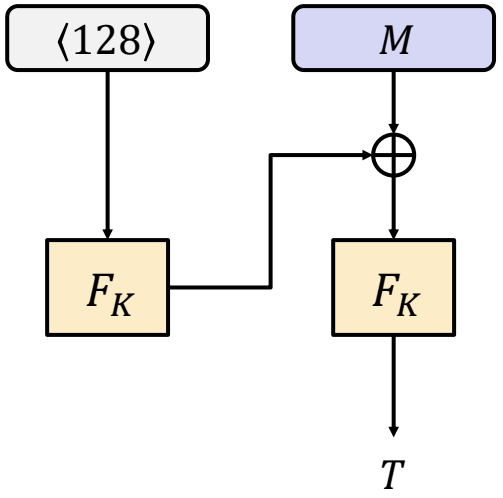
CBC-MAC

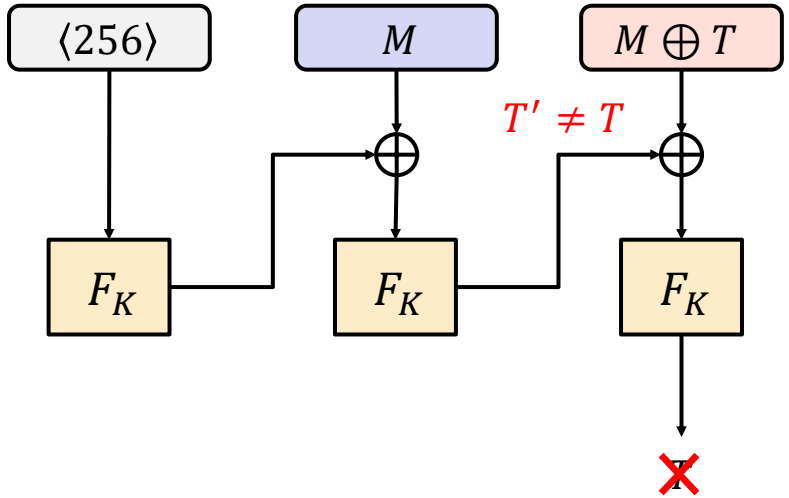




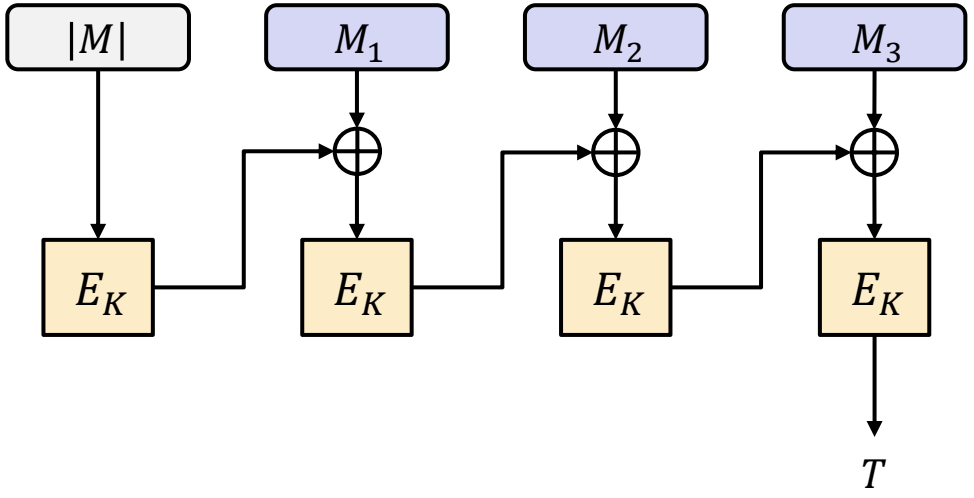
$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{CBC-MAC}}(A) \approx 1$$

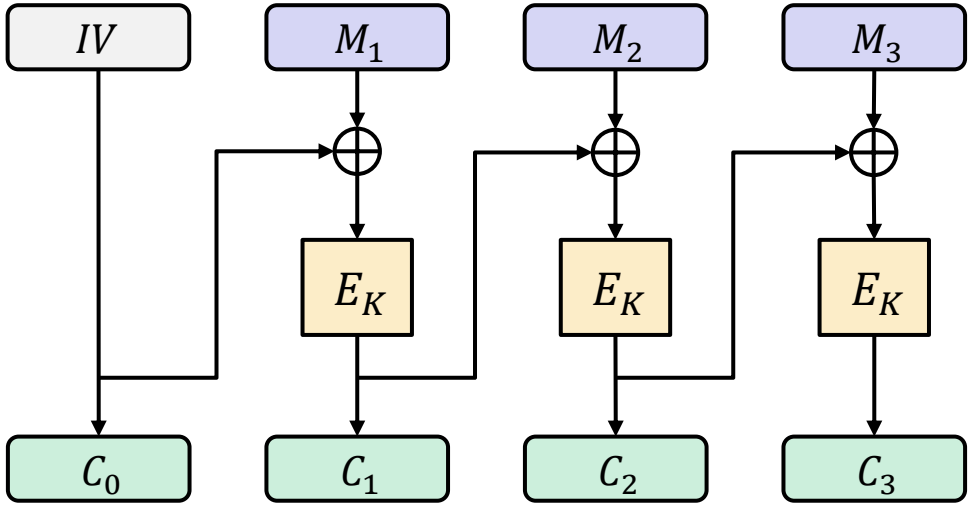
|M|-CBC-MAC





|M|-CBC-MAC vs. CBC$-ENC



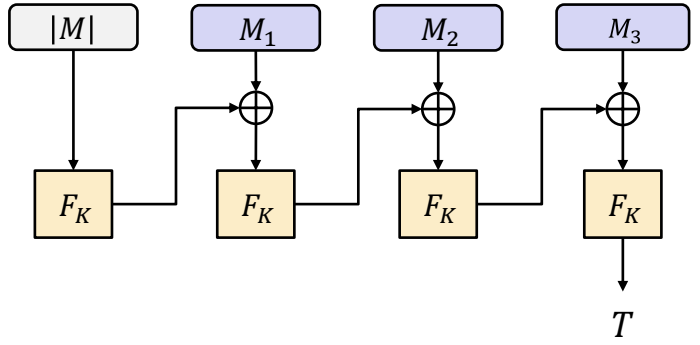


Warning 1: |M|-CBC-MAC is insecure *with* an IV!

Warning 2: CBC$-ENC is insecure *without* an IV!

Length-prepended CBC-MAC

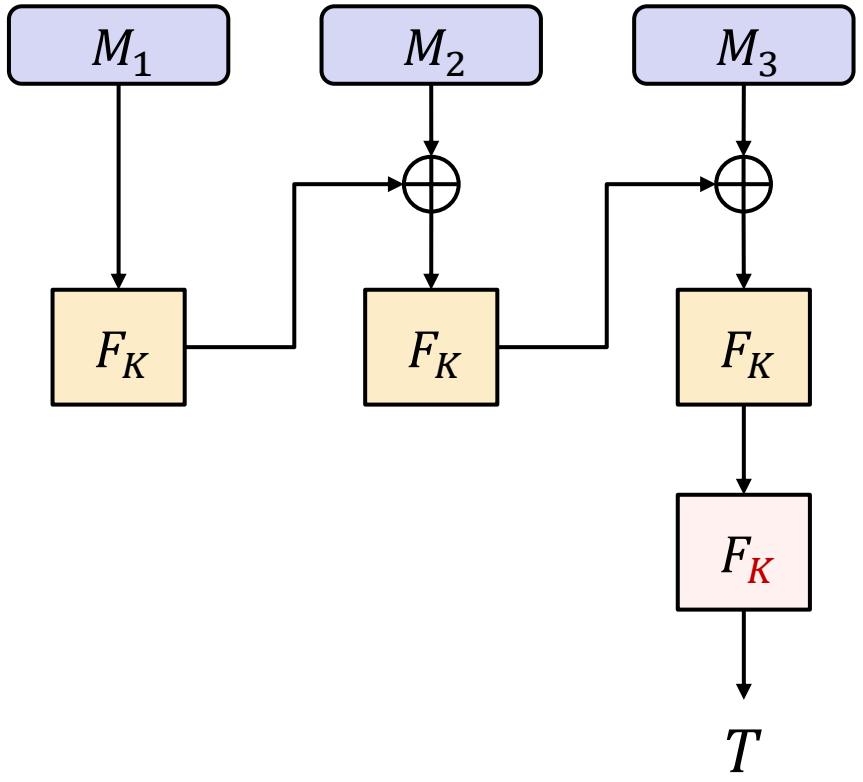
- **Theorem:** A secure variable-input-length PRF if F is a secure fixed-input-length PRF
  - Hence also UF-CMA secure



- Fully sequential
  - Cannot utilize parallel AES cores
  - Not *online* (cannot start processing before length of M is known)

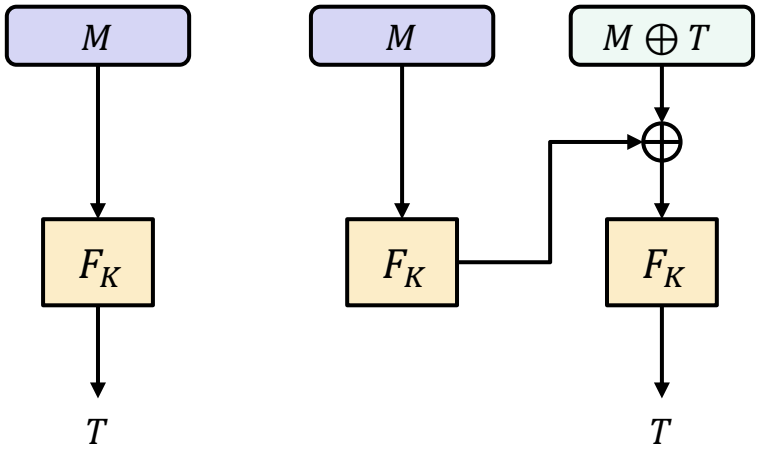
we can do better!

ECBC-MAC



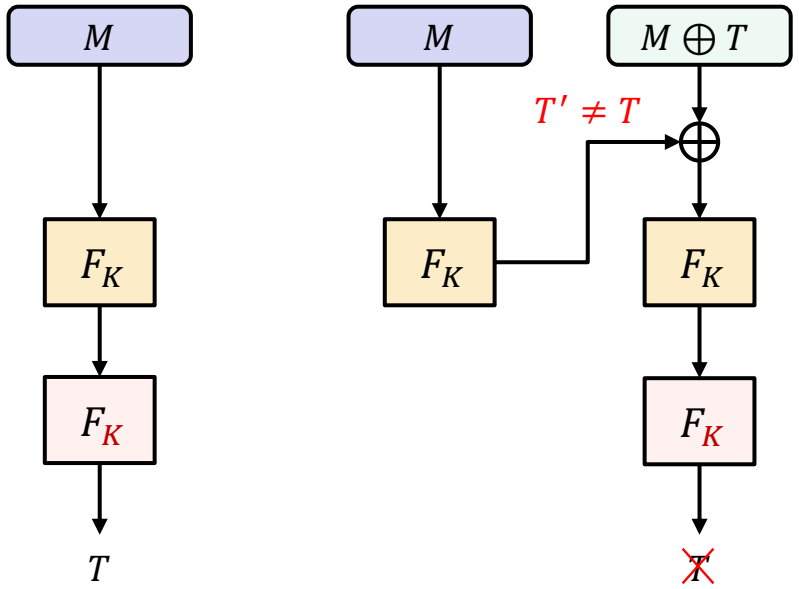
CBC-MAC vs ECBC-MAC

CBC-MAC

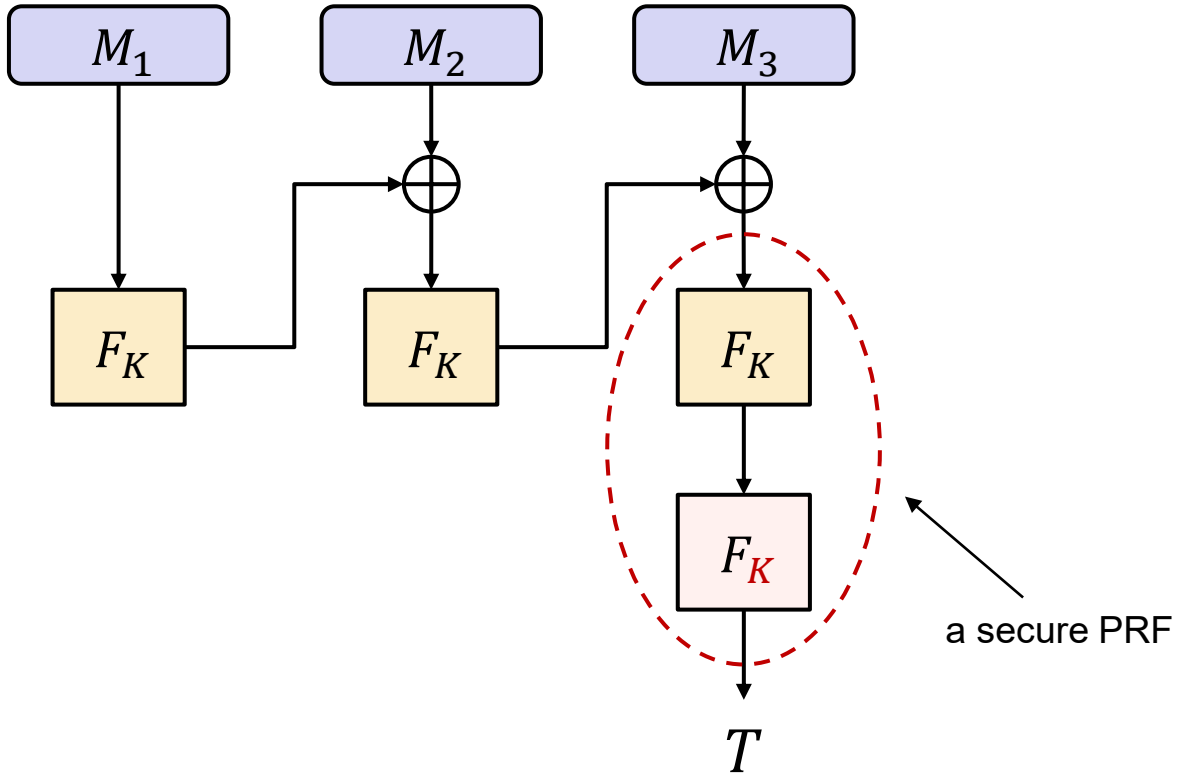


$$\mathbf{Adv}^{\mathrm{prf}}_{\mathrm{CBC-MAC}}(A) \approx 1$$

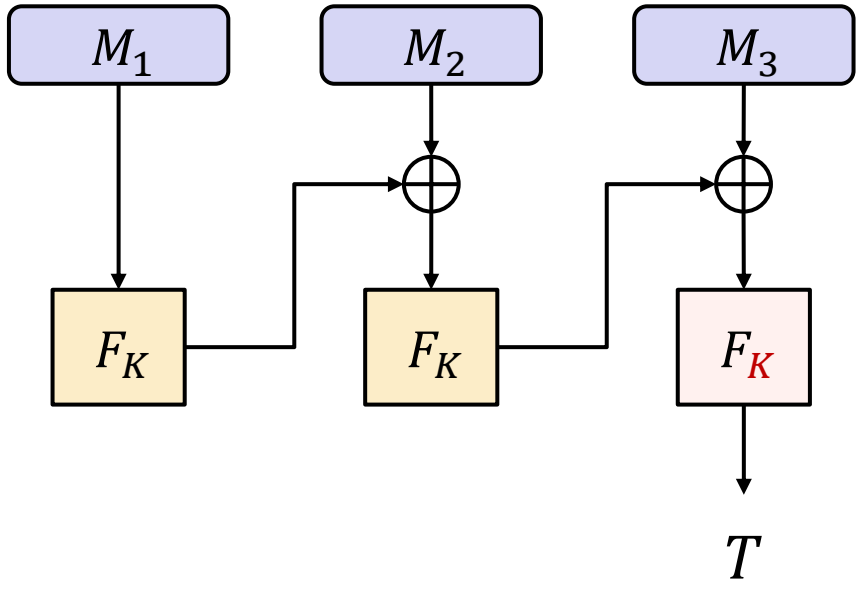
ECBC-MAC



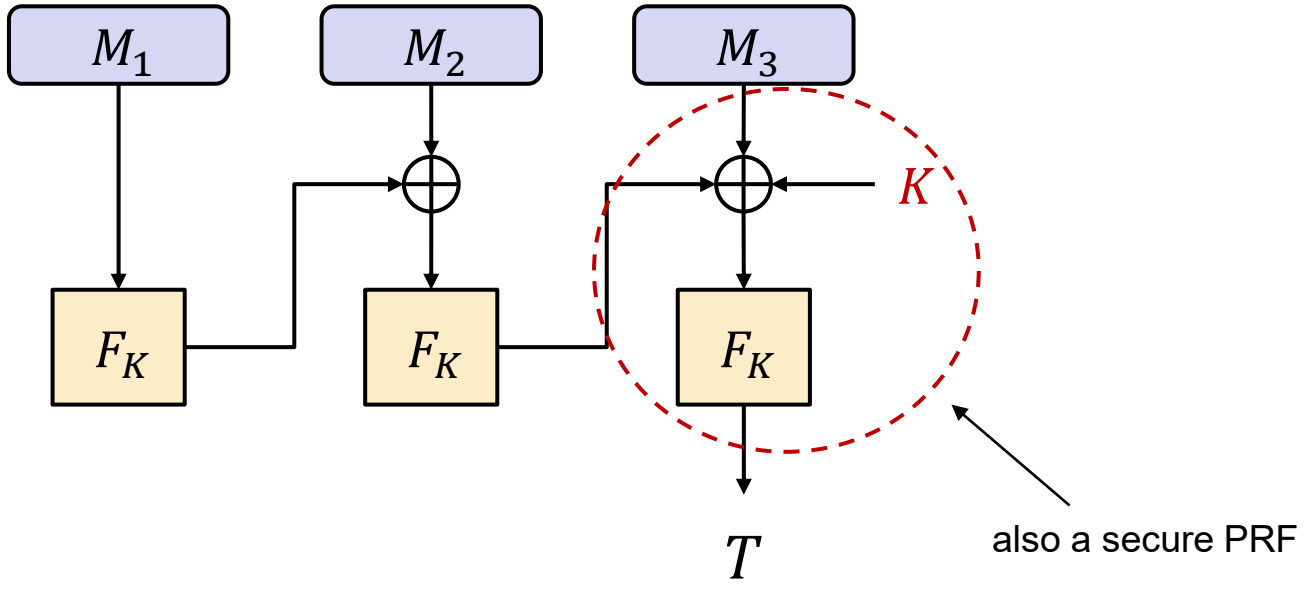
ECBC-MAC



FCBC-MAC

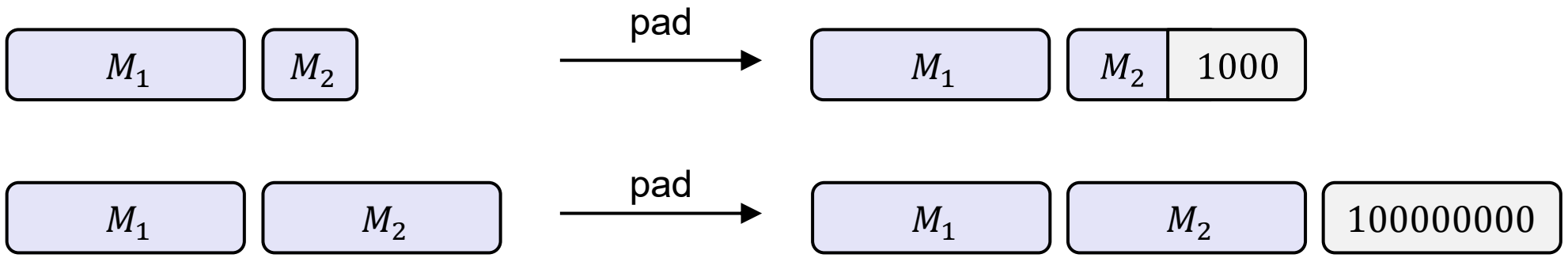


XCBC-MAC

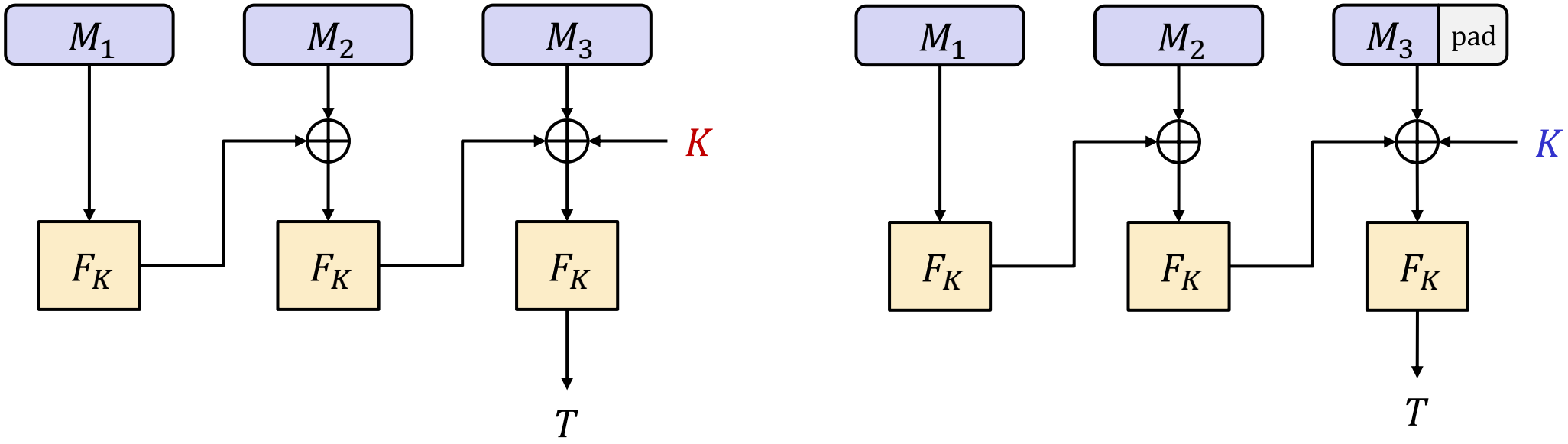


Padding

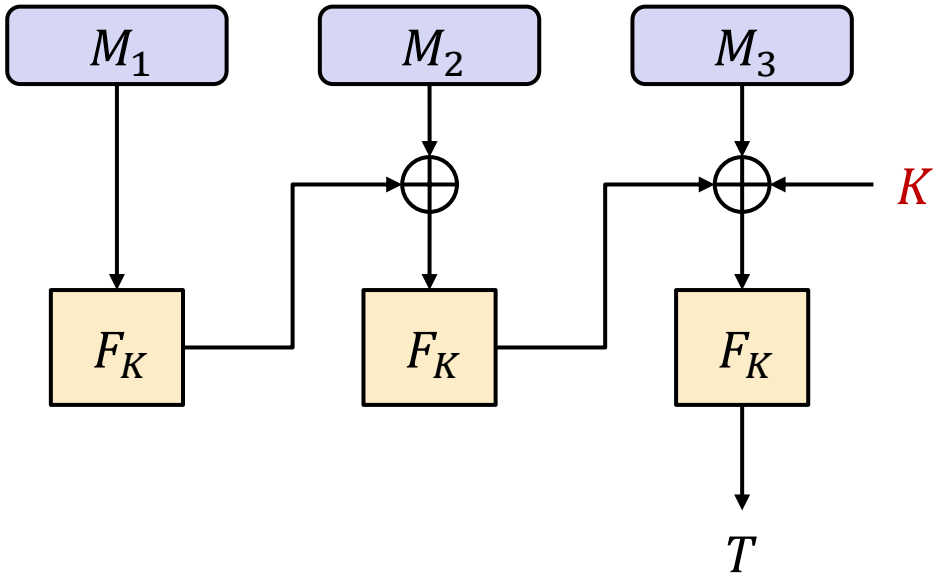
- What about messages that are not a multiple of the (fixed-input-length) PRF F's input length?
- **Padding** needed



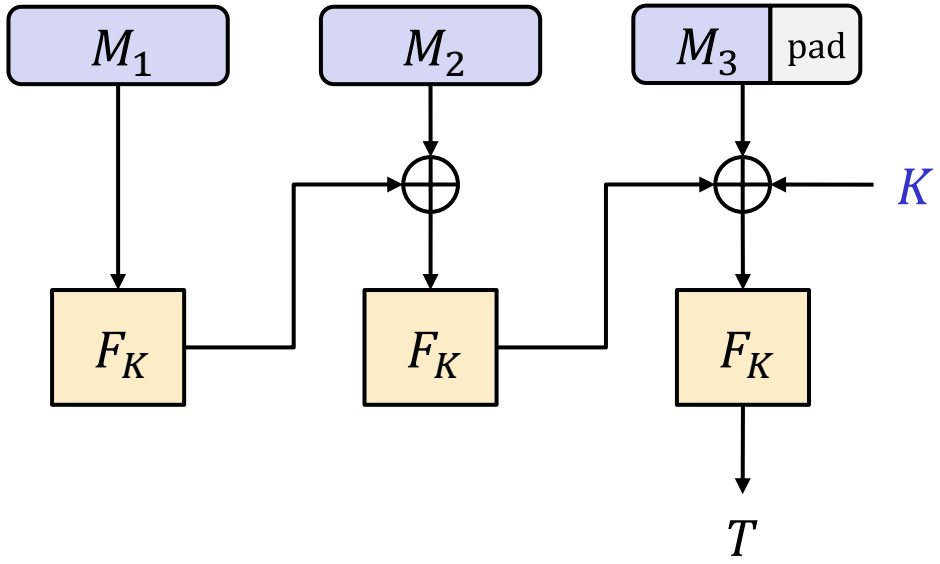
XCBC-MAC



CMAC a.k.a One-key MAC



$K = 2 * F_K(0^n)$



$K = 4 * F_K(0^n)$

Theorem: F is a secure fixed-input-length PRF $\Longrightarrow$ CMAC is a secure **variable-input-length** PRF

CMAC security

Theorem: For *any* VIL-PRF adversary A making q oracle queries (each having ℓ blocks), there is a fixed-input-length PRF adversary B such that

$$\mathbf{Adv}^{\text{vil-prf}}_{\text{CMAC}}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{(q\ell)^2}{2^n}$$

- **Example:**
  - $F = \text{AES} \quad (n = 128)$
  - Number of oracle queries: $q = 2^{40} \approx 1$ trillion queries
  - Max blocks per message: $\ell = 2^{10} \approx 16$ kB
  - $\mathbf{Adv}^{\text{prf}}_{F}(B) \approx 0$

$$\mathbf{Adv}^{\text{vil-prf}}_{\text{AES-CMAC}}(A) \leq \frac{(2^{40} \cdot 2^{10})^2}{2^{128}} = \frac{2^{100}}{2^{128}} = \frac{1}{2^{28}}$$

CMAC security

Theorem: For *any* VIL-PRF adversary A making q oracle queries (each having ℓ blocks), there is a fixed-input-length PRF adversary B such that

$$\mathbf{Adv}^{\text{vil-prf}}_{\text{CMAC}}(A) \leq \mathbf{Adv}^{\text{prf}}_{F}(B) + \frac{(q\ell)^2}{2^n}$$

- **Example:**
  - $F = \text{DES} \quad (n = 64)$
  - Number of oracle queries: $q = 2^{40} \approx 1$ trillion queries
  - Max blocks per message: $\ell = 2^{10} \approx 16$ kB
  - $\mathbf{Adv}^{\text{prf}}_{F}(B) \approx 0$

$$\mathbf{Adv}^{\text{vil-prf}}_{\text{DES-CMAC}}(A) \leq \frac{(2^{40} \cdot 2^{10})^2}{2^{64}} = \frac{2^{100}}{2^{64}} = 2^{36}$$



(since advantage is always upper bounded by 1 anyway)

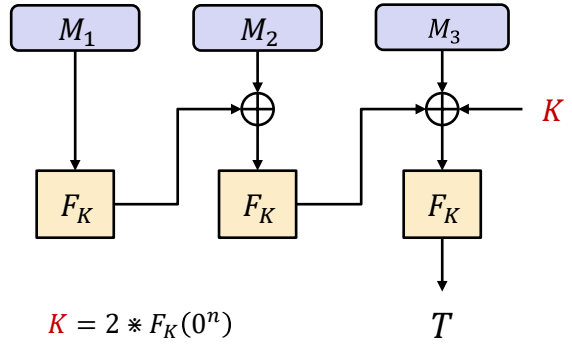
CMAC

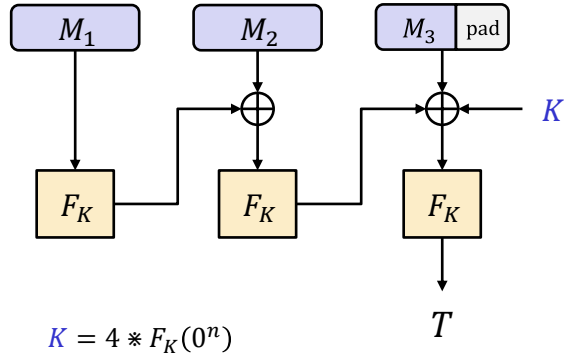
- **Theorem:** A secure variable-input-length PRF if F is a secure fixed-input-length PRF
  - Hence also UF-CMA secure

- 1 F-call per message block + 2 F-calls to derive K, K
  - Only one “master key” needed

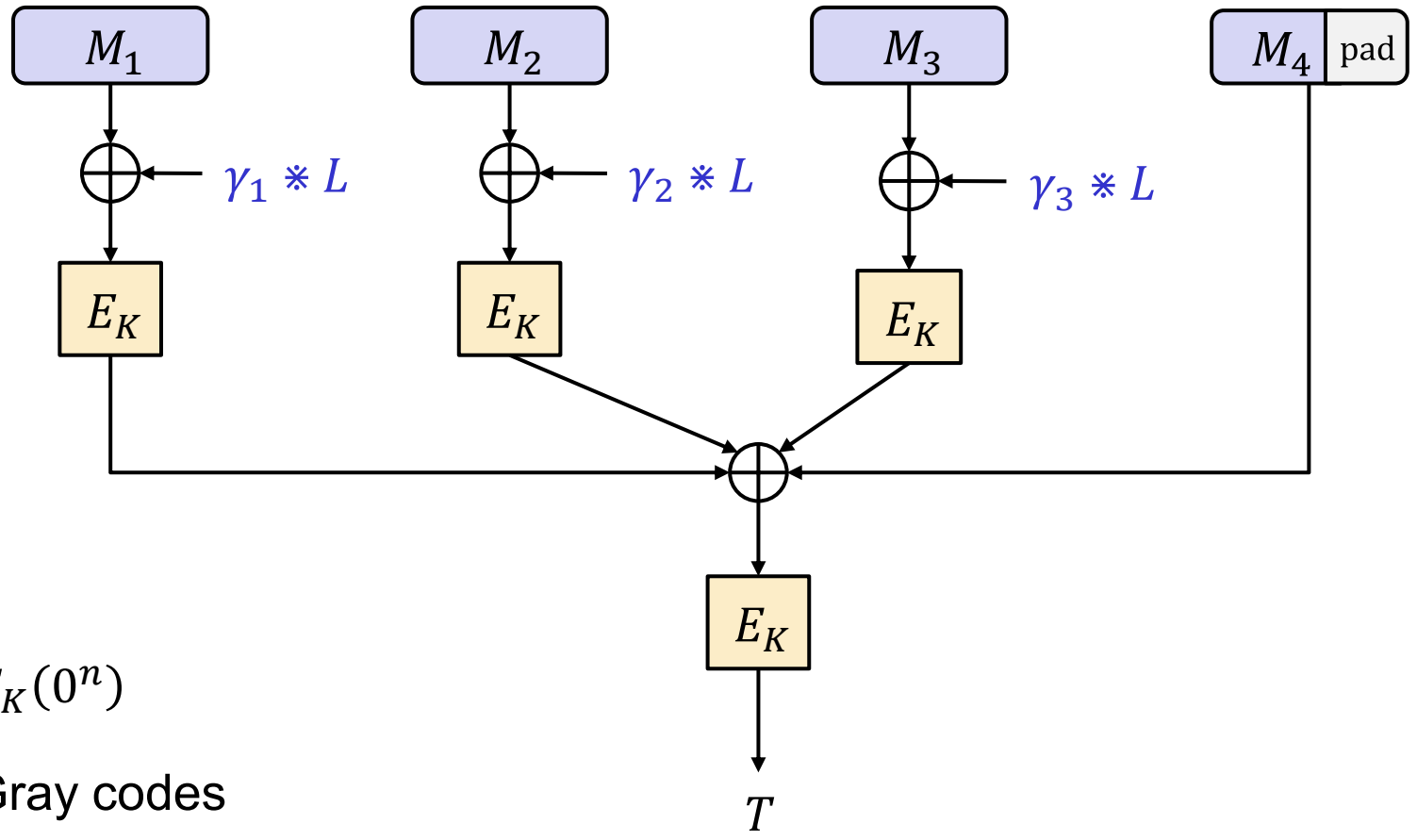
- Fully sequential
  - Cannot utilize parallel AES cores
  - But *online* (can start processing before knowing the length of M)

- Standardized by NIST
  - Widely used





PMAC



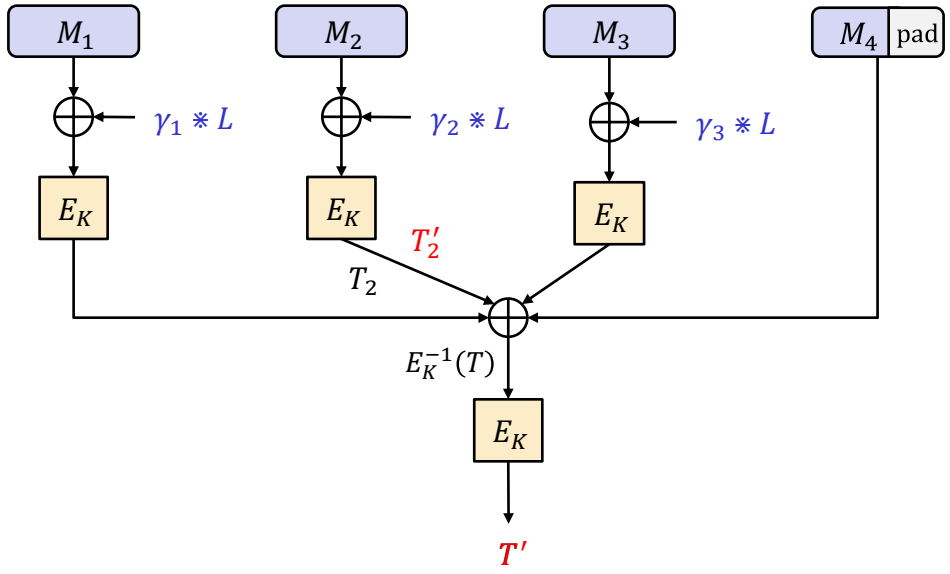
$L = E_K(0^n)$

γ_i = Gray codes

PMAC properties

- **Theorem:** A secure variable-input-length PRF if E is a secure PRP

- One key
- Fully parallelizable
- Incremental
- …not used in practice



$$\text{PMAC}_K(M_1||M_2||\ldots||M_{1000}) = T$$

$$\text{PMAC}_K(M_1||M_2'||\ldots||M_{1000}) = E_K(\,E_K^{-1}(T) \oplus T_2 \oplus T_2'\,) = T'$$

$$T_2 \leftarrow E_K(M_2 \oplus \gamma_2 * L)$$

$$T_2' \leftarrow E_K(M_2' \oplus \gamma_2 * L)$$

Summary

- UF-CMA the right security notion for message integrity
  - Does not cover replay attacks

- Variable-length input PRFs are good MACs
  - Hence only need to focus on creating secure VIL-PRFs

- CBC-MAC good PRF for messages of a ***single*** *fixed* input length

- CMAC upgrades CBC-MAC to handle all input lengths